

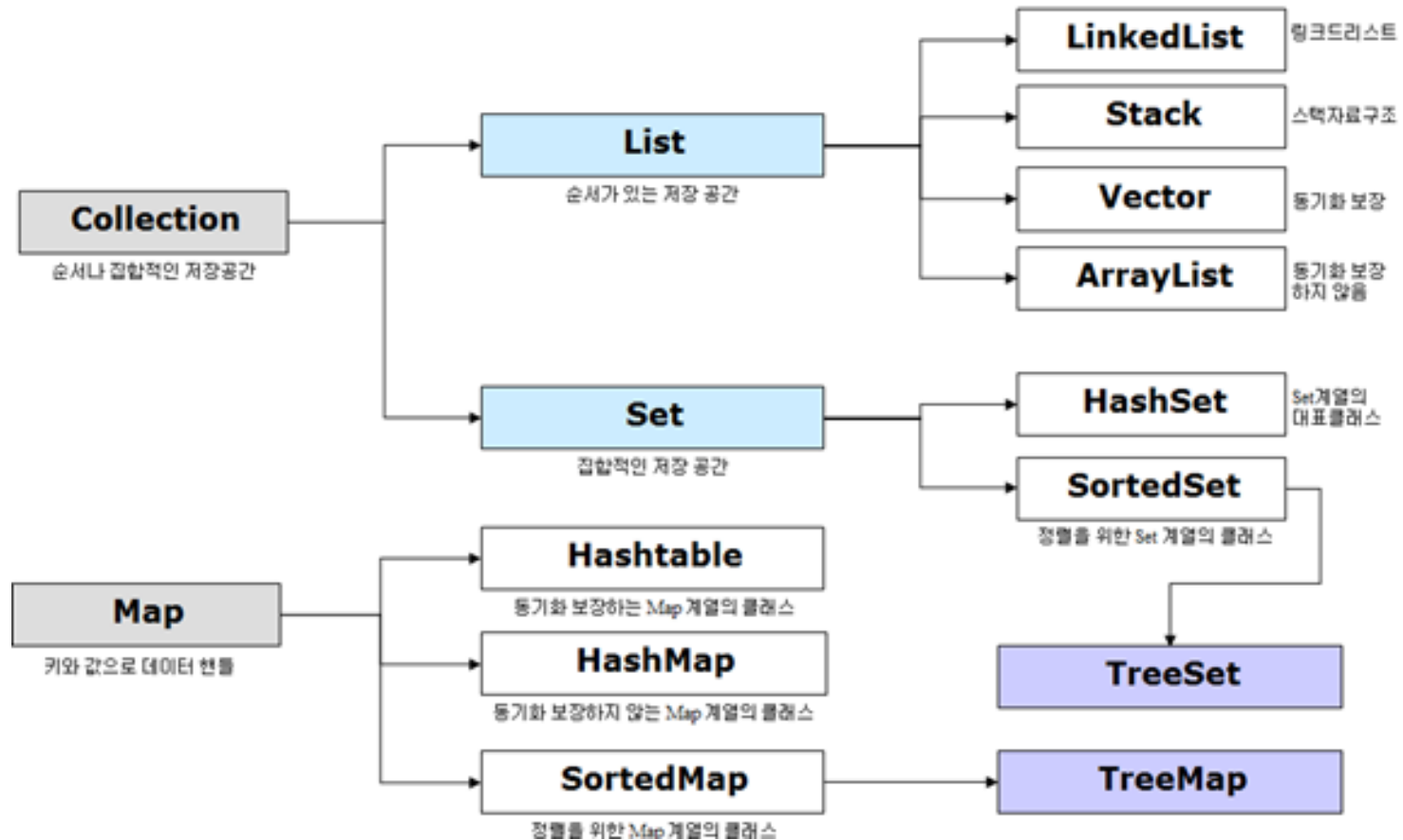
# JAVA Collection & MAP

2017. 05. 17.

김상영

# JAVA Collection Framework

## ● JAVA Collection Framework의 상속 기본구조



# Collection

## ● Collection

- List, Set Interface는 Collection Interface를 상속.

## ● List

- 순서가 있는 Collection. (이 순서는 삽입된 순서를 의미)
- Data를 중복해서 포함할 수 있음.
- List의 종류: LinkedList, Stack, Vector, ArrayList.

## ● Set

- 집합적인 개념의 Collection. (순서의 의미가 없다)
- Data를 중복해서 포함할 수 없음.
- Set의 종류: HashSet, SortedSet.

# List

## ● Stack

- Data의 삽입과 추출이 후입선출(Last-In First-Out) 구조.

## ● Vector

- 자동으로 동기화를 보장.
- ArrayList에 동기화가 보장되도록 최적화한 클래스.
- JAVA 5.0 이후로는 AutoBoxing/AutoUnBoxing을 지원.

## ● ArrayList

- 동기화를 보장해주지 않음.
- 배열에 동적 메모리 증가 기능을 구현한 클래스.

# Set

## ● HashSet

- 가장 빠른 임의 접근 속도.
- 순서를 전혀 예측할 수 없음.

## ● LinkedHashMap

- 추가된 순서, 또는 가장 최근에 접근한 순서대로 접근 가능.

## ● TreeSet

- Set 개념에 정렬 기능이 가미된 클래스.
- 정렬된 순서대로 보관하며 정렬 방법을 지정할 수 있음.
- 데이터들이 자동으로 오름차순으로 정렬.

# Map

## ● Map

- 기존 List, Set Interface의 검색의 개념이 추가된 Interface.
- 데이터 삽입시 <Key, Value> 형태로 삽입되며 Key를 이용해 Value를 반환.

## ● Hashtable

- HashMap보다는 느리지만 동기화가 지원
- 키와 값으로 NULL이 허용되지 않음.

## ● HashMap

- Map 인터페이스를 구현하기 위해 해시테이블을 사용한 클래스.
- 중복을 허용하지 않고 순서를 보장하지 않음
- 키와 값으로 NULL이 허용

# Map

## ● TreeMap

- 이진검색트리의 형태로 키와 값의 쌍으로 이루어진 데이터를 저장.
- 정렬된 순서로 키/값 쌍을 저장하므로 빠른 검색이 가능.
- 저장시 Key 기준정렬(오름차순)을 하기 때문에 저장시간이 다소 오래 걸림.
- Map 인터페이스를 상속한 SortedMap 인터페이스를 구현한 클래스.

## ● LinkedHashMap

- 기본적으로 HashMap을 상속받아 HashMap과 매우 흡사.
- Map에 있는 엔트리들의 연결 리스트를 유지하므로 입력한 순서대로 반복 가능

# Comparator

## ● Comparator

- TreeSet과 TreeMap은 사용자가 직접 정렬 방식을 지정할 수 있음.
- TreeSet과 TreeMap은 정렬을 위한 Comparator 인터페이스를 구현.
- TreeSet에 Data를 집어넣으면 기본적으로 오름차순 정렬이 된다.
- 기본 정렬기준은 단순한 문자열이나 기본 데이터 타입과 같은 대상에만 적용됨.
- 이에 사용자가 직접 비교법을 넣어주기 위해 사용하는 것이 Comparator.



# Map

## ● Hashtable, HashMap 공통점

- 내부적으로 모두 Hash 기법을 이용.
- Map 인터페이스를 구현.
- Key, Value를 이용해서 Data를 관리.

## ● Hashtable, HashMap 차이점

- Hashtable은 동기화가 보장.
- HashMap은 동기화가 보장되지 않음.

## ● Hashtable, HashMap과 HashSet과의 관계

- Hashtable과 HashMap은 둘 다 Map 인터페이스를 구현.
- HashSet은 내부적으로 Hash 기법을 사용하나 Set 인터페이스를 구현.